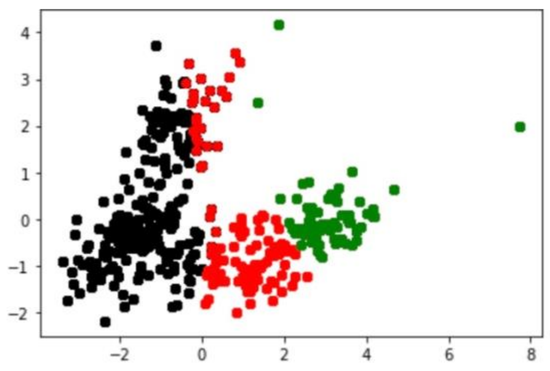
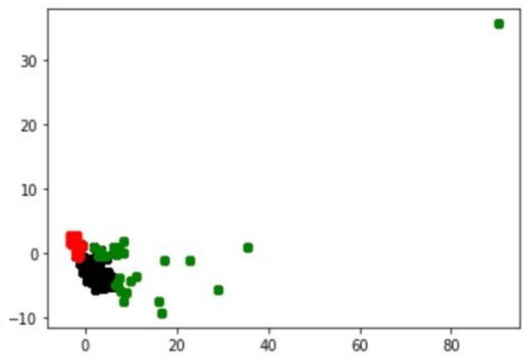
Implementation Summary

I began with loading in both the Cho and Iyer datasets as data frames using pandas. I downloaded them into my local directory I was using Jupyter Notebooks in. For data cleaning, I removed outliers which were designated as -1 values in the ground_truth attributes for cho and iyer. Next, I checked to see if there were duplicates in both datasets, which I found none existing. Next, I dropped the gene_id column from the dataset because I didn't want the id's to interfere with my clustering algorithms. Next, I performed PCA Reduction on the datasets. So that I could later graph them on a scatter plot, I reduced both datasets to two dimensions. I tested the K-Means algorithm several times by printing the generated centroids and also printing the elements in their clusters to ensure the elements from each dataset were assigned a cluster correctly and that these clusters were being recomputed based on the mean of the values in the cluster.

My results from the k-means clustering are shown below where I used k = 3 clusters. I was surprised to see the outlier in Iyer because I had removed all outliers identified by their ground truth value as -1.



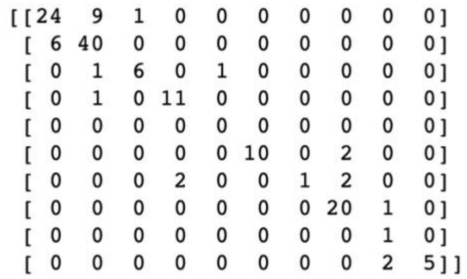
Cho Dataset



Iyer Dataset

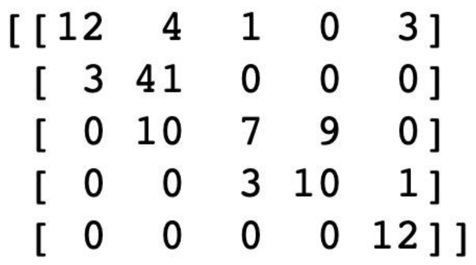
I had a difficult time generating the scatter plots for my K-Means algorithm. I initially used a dictionary to store an array of the elements as the value where the key was the cluster they were assigned to, however, accessing this dictionary and assigning colors to these keys generated many errors. Intitially, the scatter plot only generated two colors while I had created three that corresponded to my three clusters. After troubleshooting I found indexing issues existed at the time of plotting, and I added a list of lists allowing indexing by converting given clusters into tuples.

My implementation of spectral clustering involved four steps. First I represented the data points as a symmetric similarity graph. Doing so I chose to use the k-NN algorithm using the KNeighborsClassifier().



```
[[24  9  1  0  0  0  0  0  0  0]
 [ 6 40  0  0  0  0  0  0  0  0]
 [ 0  1  6  0  1  0  0  0  0  0]
 [ 0  1  0 11  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0 10  0  2  0  0]
 [ 0  0  0  2  0  0  1  2  0  0]
 [ 0  0  0  0  0  0  0 20  1  0]
 [ 0  0  0  0  0  0  0  0  1  0]
 [ 0  0  0  0  0  0  0  0  2  5]]
```

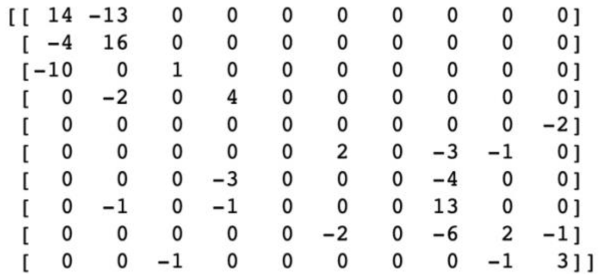
Iyer Symmetric Similarity Graph



```
[[12  4  1  0  3]
 [ 3 41  0  0  0]
 [ 0 10  7  9  0]
 [ 0  0  3 10  1]
 [ 0  0  0  0 12]]
```

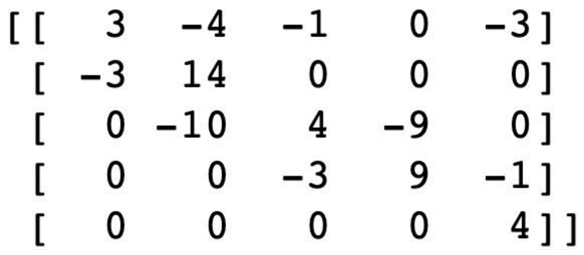
Cho Symmetric Similarity Graph

Next, I compute the graph Laplacian for both datasets using np.linalg.eig().



```
[[ 14 -13   0   0   0   0   0   0   0   0]
 [ -4  16   0   0   0   0   0   0   0   0]
 [-10   0   1   0   0   0   0   0   0   0]
 [  0  -2   0   4   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0  -2]
 [  0   0   0   0   0   2   0  -3  -1   0]
 [  0   0   0  -3   0   0   0  -4   0   0]
 [  0  -1   0  -1   0   0   0  13   0   0]
 [  0   0   0   0   0  -2   0  -6   2  -1]
 [  0   0  -1   0   0   0   0   0  -1   3]]
```

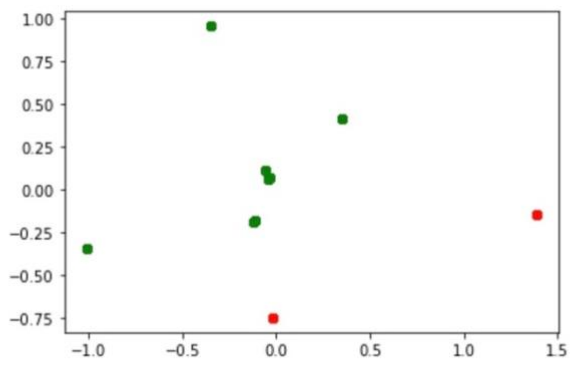
Iyer Graph Laplacian



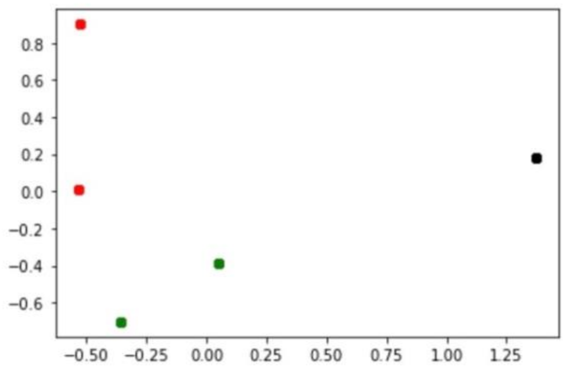
```
[[  3  -4  -1   0  -3]
 [ -3  14   0   0   0]
 [  0 -10   4  -9   0]
 [  0   0  -3   9  -1]
 [  0   0   0   0   4]]
```

Cho Graph Laplacian

After this step, I computed K eigenvectors corresponding to the k smallest non-zero eigenvalues of L. This step generated a eigenvector matrix for both datasets and passed these eigenvectors into my k-means algorithm. The best result I generated with my spectral clustering implementation is shown below.



Iyer Dataset



Cho Dataset

My work concluded the K-Means algorithm was much simpler to implement than the Spectral clustering algorithm. K-Means returned what appeared to be the more reliable cluster. I found that PCA reduction was an extremely useful task in plotting my results. I conclude the Iyer dataset showed better clustering than the Cho dataset with both algorithms used. While the Iyer dataset did have one outlier, even after I removed the outliers, it still seemed to have closer and less noisy clusters than the Cho dataset had after performing the algorithms on the preprocessed data.